The Automated Satellite Data Processing System

AVHRR Processing

Naval Research Laboratory, Stennis Space Center, MS
Paul Martinolich
Theresa Scardino

The Automated Satellite Data Processing System: AVHRR Processing

by Naval Research Laboratory, Stennis Space Center, MS, Paul Martinolich, and Theresa Scardino

Publication date 18 November 2009

Table of Contents

List of Figures

List of Tables

List of Examples

List of Equations

Part I. Advanced Very High Resolution Radiometer

The chapters in Part I form a User's Guide for the processing of the Advanced Very High Resolution Radiometer (AVHRR) within the Automated Processing System. The reader should be familar with the overall concept of the Automated Processing System (*APS User's Guide*).

Chapter 1. Introduction

The Advanced Very High Resolution Radiometer (AVHRR) is a space-borne sensor embarked on the National Oceanic and Atmospheric Administration (NOAA) family of polar orbiting platforms (POES). AVHRR instruments measure the reflectance of the Earth in up to 6 relatively wide (by today's standards) spectral bands. The first two bands are centred around the red (0.6 micrometer) and near-infrared (0.9 micrometer) regions. In the short-wave-infrared regions, two bands sample the 1.6 and 3.8 micrometer regions. These bands are given the designations 3A and 3B. The last two sample the thermal radiation emitted by the planet, around 11 and 12 micrometers, respectively. They are commonly designationed band 4 and 5.

The first AVHRR instrument actually was a 4-channel radiometer (bands 1, 2, 3B, and 4). The second version of the AVHRR instrument added the 12 micrometer band. The latest version (known as AVHRR/3, first carried on the NOAA-15 platform launched in May 1998) acquires data in a 6th channel located at 1.6 micrometer. Due to the original design of the transmission format which allowed for only five bands, the new SWIR band replaces the original 3.8 micrometer band during day time viewing.

Chapter 2. AVHRR Products

In the ocean community, the AVHRR sensor is the primary source of sea surface temperature data. It has a stable platform for that purpose over several decades. And, following on the original work of Stumpf (1992), the AVHRR data has yielded some other useful ocean parameters, like, beam attenuation, though the usefulness of this data is more challenging given the greater difficulty in atmospheric correction.

Thermal Channel Products

The AVHRR has two channels in the long-wave radiation of the electro-magnetic spectrum at 10 um and 11 um which are used to produce estimates of the sea surface temperature during both day and night. A third channel at 3.9 um is useful for night-time sea surface temperature. The algorithms used to produce these sea surface temperature estimates include the multi-channel sea surface temperature (MCSST) temperature and non-linear sea surface temperature (NLSST). See avhIngest(1) and avhSST(1) for details.

Figure 2.1. Brightness Temperature 3.9 um Product

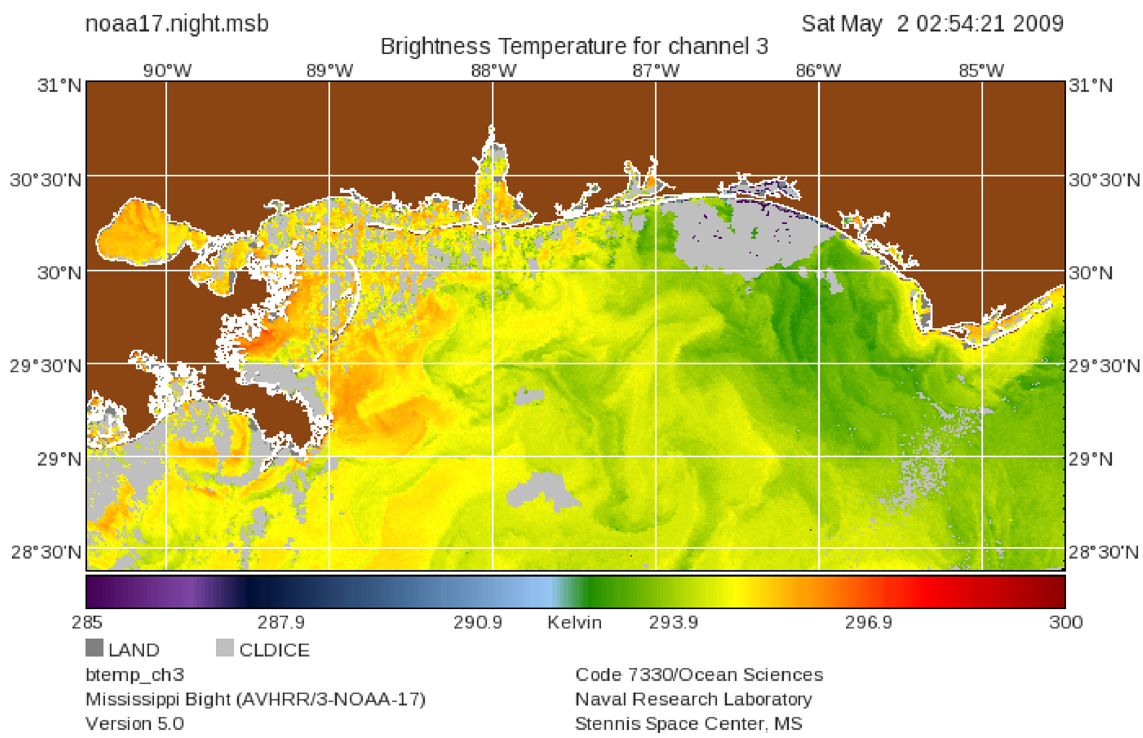


Figure 2.2. Brightness Temperature 10 um Product

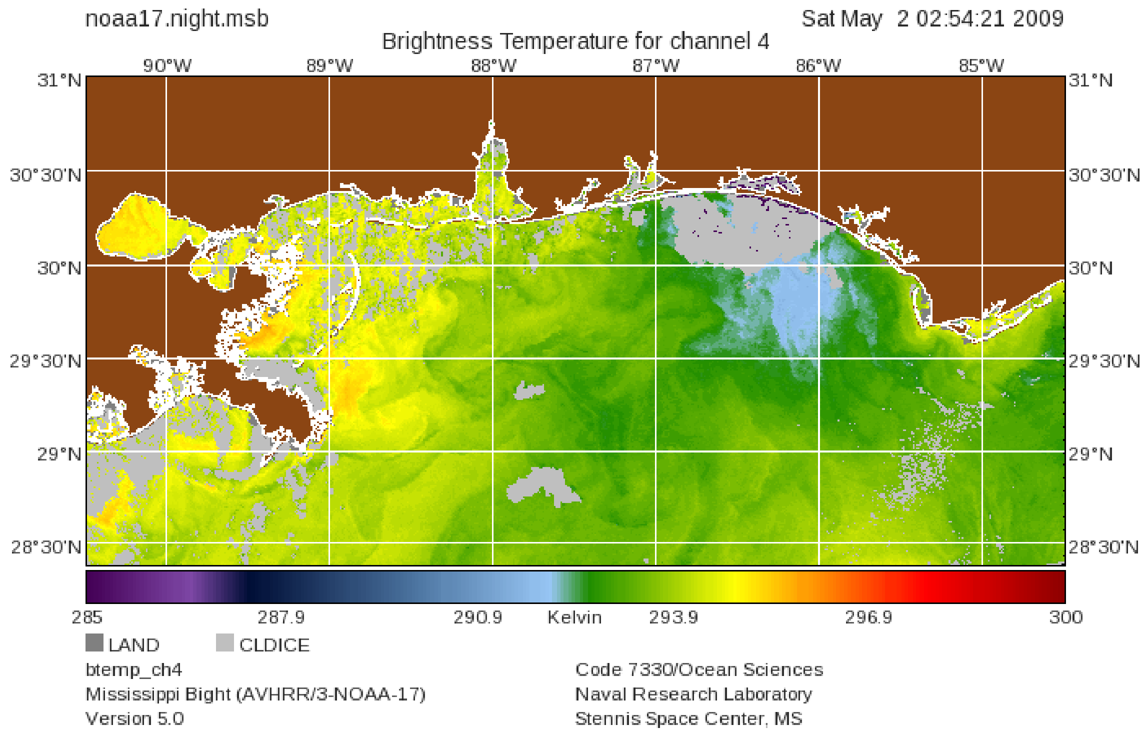


Figure 2.3. Brightness Temperature 11 um Product

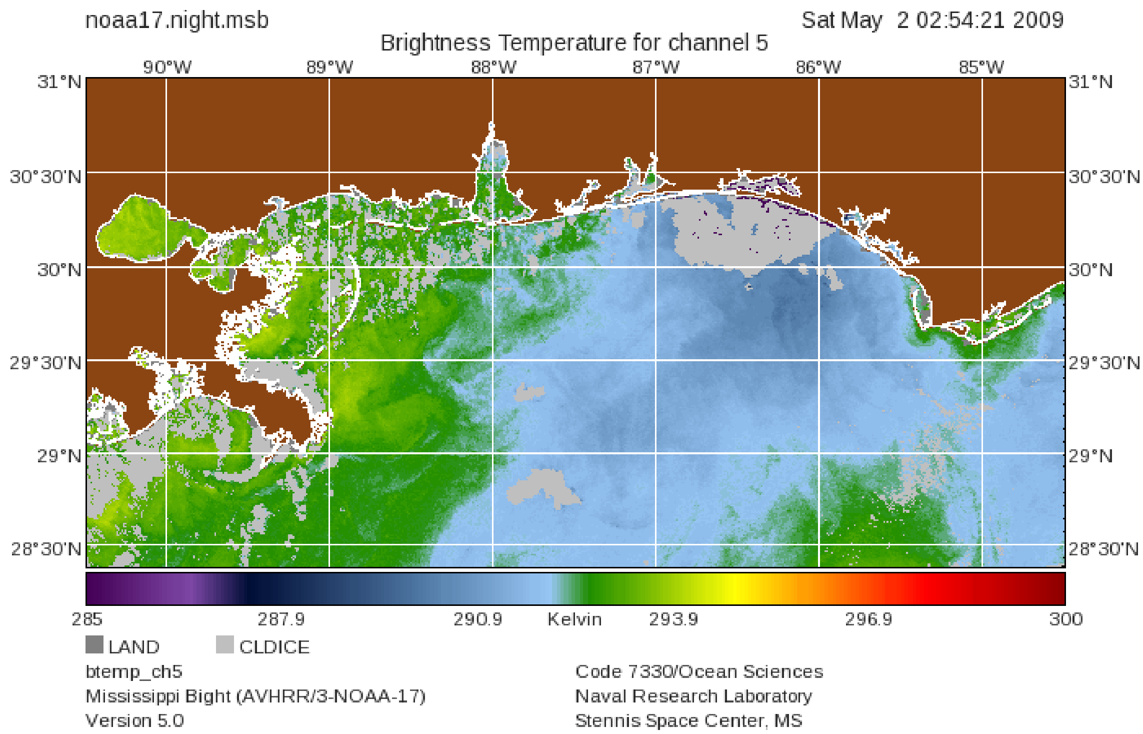
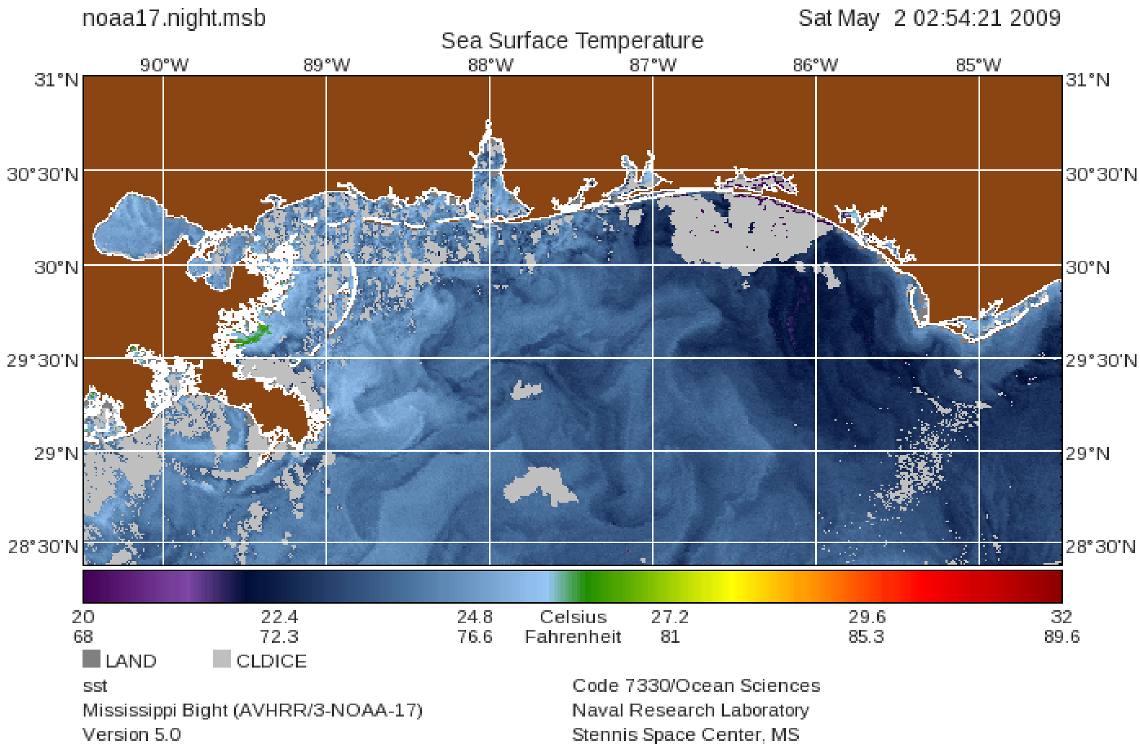


Figure 2.4. Sea Surface Temperature Product



Visible Channel Products

The AVHRR has two channels in the visible and near-ir region of the electro-magnetic spectrum. Using the algorithms based on the work of Stumpf, R. P. (1992) *Remote Sensing of Water Clarity and Suspended Sediments in Coastal Waters*, estimates of several optical fields can be generated. The images below show examples of beam attenuation, K_PAR, and suspended sedimates. See avhTurbid(1) for details.

Figure 2.5. Albedo Channel 1 Product

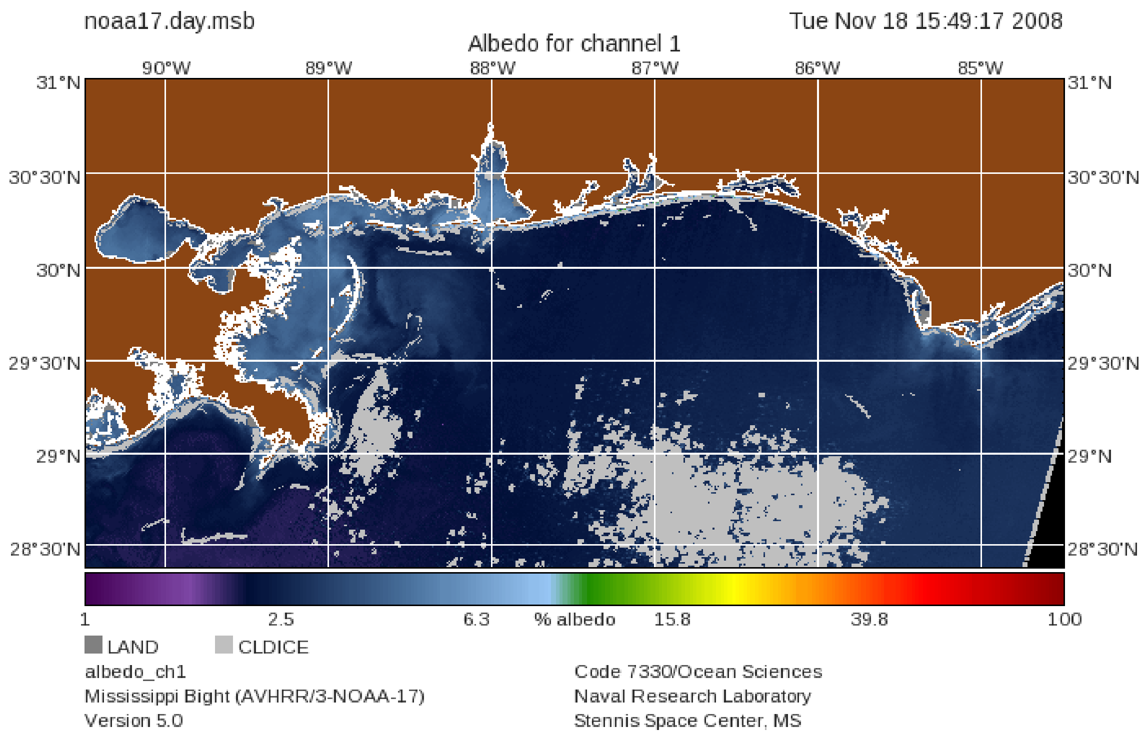


Figure 2.6. Beam Attenuation Product

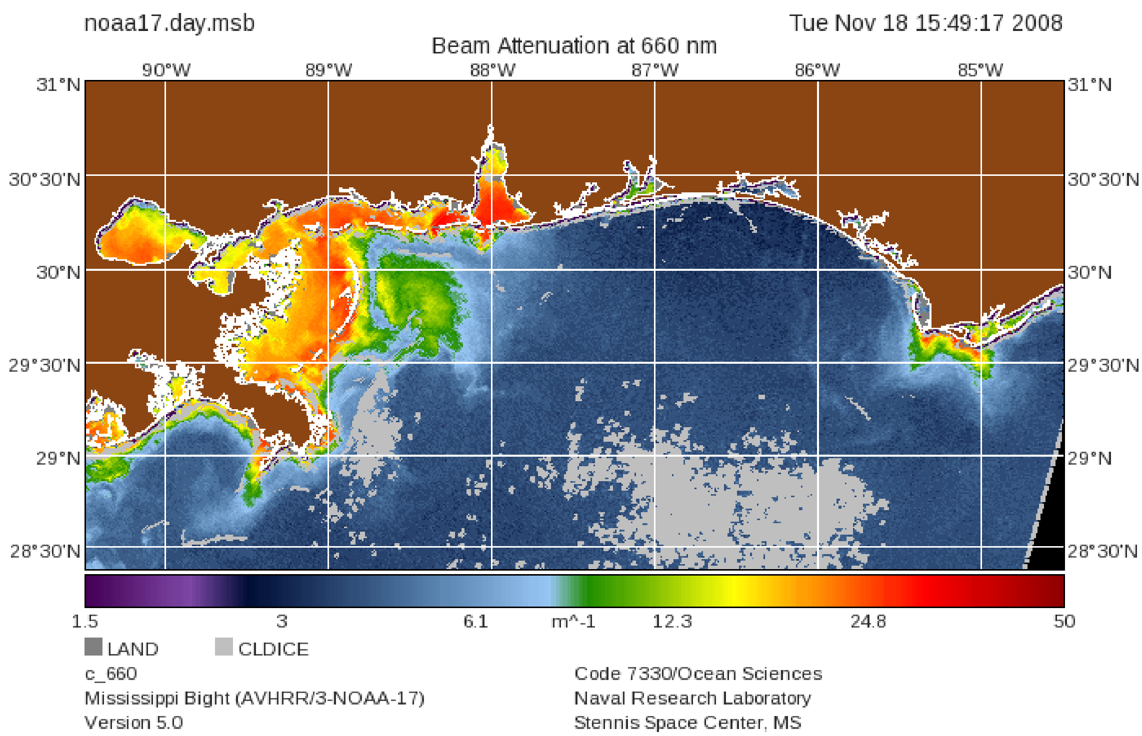


Figure 2.7. K(PAR) Product

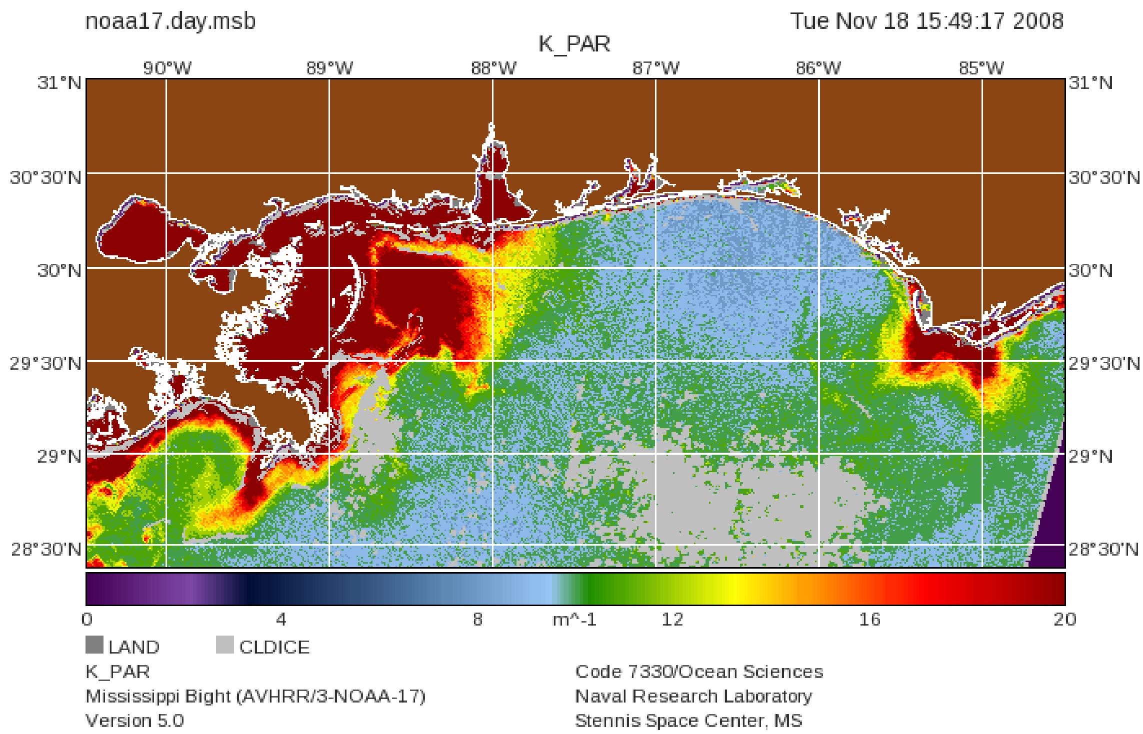
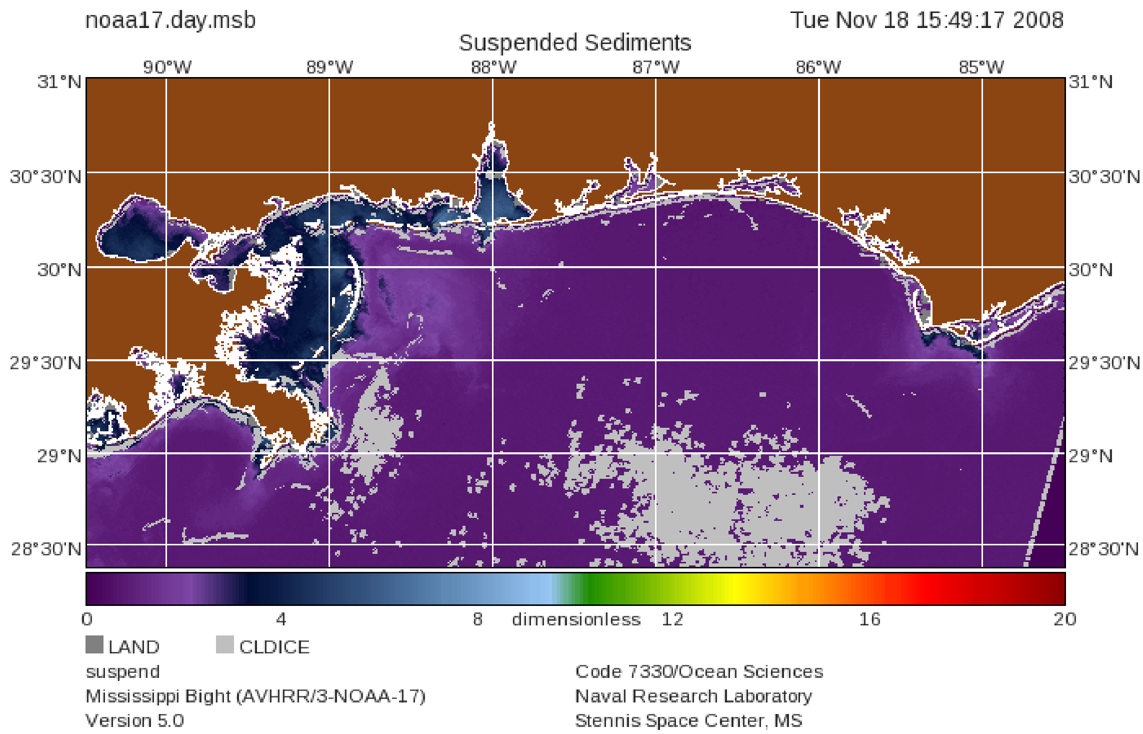


Figure 2.8. Suspended Solids Product



Processing Flags

The AVHRR data includes a product that contains processing flags for each pixel. These flags are used for masking data and other information data. For example, one flag holds an indication of possible fire. When the fire detection algorithm determines that the given pixel, this bit is set to one.

Data Input Formats

The AVHRR data can be processed from the following formats: NESDIS Level-1B format, NESDIS Level-1B KLM (all versions), and an NRL Terascan/HDF format. Some APS programs can query the NESDIS Level-0 format, however, no calibrated data can be generated. Several of the APS programs take the AVHRR data as input, including: **avhArea**, **avhDump**, **avhImage**, **avhIngest**, and **avhScan**.

The script used to create the NRL TeraScan/HDF format is included in the distribution under the share/avhrr directory. The script uses the TeraScan commands **hrptin**, **nav**, and **angles** to create a TeraScan data file of calibrated radiance data with geometery angles and earth location. The final step is to convert the file from the TDF format to a more general HDF format using the TeraScan software **tdf2hdf**.

Data Output Formats

The AVHRR output data will be in an APS formatted file. The APS format can be written using HDF Version 4 (`hdf`), HDF Version 55 (`h5`). Due to conflicts with the use of an UNLIMITED data set, the netCDF Version 3 format (`nc`) is not available. See the *APS User's Guide* for details about the APS format.

The base format of the APS data file is usually selected by the file extension (as shown above). Those programs that use the APS format also provide an option to select the format using a switch.

Chapter 3. Standard Processing

The AVHRR standard processing under the Automate Processing System (APS) is controlled primarily by the scripts functions located in the file `avhScripts.sh`. This file can be found in the APS library directory `APS_LIB`. The script function `avhProcess` is the main entry point which should be called by the AVHRR areas scripts. These steps are:

AVHRR Standard Processing Tasks

1. Verify input file is a known AVHRR file format using filefmt program.
2. Determine if file covers user defined map using avhArea program.
3. Get time information from file using avhInfo program.
4. Parse `$L3ProdList` to determine which intermediate products are required.
5. Ingest AVHRR data and calibrate it using avhIngest program.
6. If user selected “sst” as a product, generate it from brightness temperatures using avhSST program.
7. Generate the cloud masks and add to “l2_flags” using avhClouds program.
8. If user selected any of the visible data products and input is day scene, generate them using the avhTurbid program.
9. Optionally create Level-2 browse images.
10. Warp all the products generated above to the defined map projection.
11. For a large product array, tile and compress the data.
12. Optionally archive Level-2 datafile.
13. Generate statics on Level-3 datafile using imgStat (if available).
14. If user has defined `$AvhPreBrowse`, it is now called. (Normally, not defined.)
15. If user had defined `$L3BrowseList`, generate browse images for selected products.
16. If user has defined `$AvhPostProcess`, it is now called. (Normally, not defined.)
17. Optionally, extract time series and point information.
18. Store Level-3 file in the `$APS_DATA_BASE` and the browse image in the `$APS_IMAG_BASE`.
19. If user set `$L4ProdList`, then run daily, 8-day, monthly, and yearly composites.
20. Optionally, run any user post-processing functions.

Usage

A “minimalist” executable script for processing AVHRR data must source both the apsScripts and avhScripts located in the `$APS_LIB/aps` directory, define a required variable and call the script function avhProcess passing it the name of the input AVHRR file. Many other variables can be optionally set to modify the "normal" mode of operations. These must be set prior to the call to avhProcess.

```
#!/bin/bash
 . $APS_LIB/aps/apsScripts.sh
 . $APS_LIB/aps/avhScripts.sh
MapName=GulfOfMexico
avhProcess $1 $0
```

This script must be placed in the `$AREAS_PROC` directory which is normally the directory `areas` located in the main APS directory. The script must have execution permissions.

Required Variables

These variables are required to process an area. They provide the script an area to process. Most of the remaining variables have defaults which can be overridden. They are described in the next section.

`MapName` This is the name of the image map stored in the file $MapFile.

Optional Variables

The following sections are variables that have defaults which the user can override to change the behaviour of the default processing. They are grouped together by subject.

Product Selection

`L3ProdList` This is a space delimited list of products to be written to the output data base. It is initially defined by APS, through the `$APS_ETC/avhrr_defaults.sh` file. The following products are available:

Table 3.1. List of available AVHRR products

Name	Description
albedo_ch1	Percent albedo from channel 1
albedo_ch2	Percent albedo from channel 2
albedo_ch3	Percent albedo from channel 3A
btemp_ch3	Brightness temperature from channel 3
btemp_ch4	Brightness temperature from channel 4
btemp_ch5	Brightness temperature from channel 5
sst	Sea surface temperature
c_660	Beam attenuation at 660 nm.
K_PAR	Diffuse Attenuation for photosynthetically active radiation
suspend	Total suspened solids concentration
Ray_ch1	Rayleigh reflectance for channel 1
ref_dif	Reflectance difference between channel 1 and channel 2

If not defined, the default value is "sst clouds c_660". For example, the user might put the line

```
L3ProdList="btemp_ch3 btemp_ch4 btemp_ch5 sst"
```

to retain only the thermal channel information.

Data Base

These variables control information about where the data base of Level-3 and Level-4 will reside and the structure of that data base.

`DataFormat`
: This variable will be used to select the base format for the APS data files. The optional values include: `hdf`, `h5`, and `nc`.

`Region`
: This variable will be used to create the default data base directories. By default it set to `$MapName`.

`L3DataBase`
: This variable is used to indicate the location of the image data base for the generated product file. By default, it is set to:

```
$ApsDataBase/$Level/$Sensor/$Version/$Region/
$Year/$Month
```

where, `$ApsDataBase` is defined in the `aps.conf` file and represents the top directory of the data base. `$Level` is set to the string "lvl3" by `avhInit`. `$Sensor` is set to the string "avhrr" by `avhInit`. `$Version` is set to "5.0" by `avhInit`. `$Region` is set to `$MapName` by `avhInit`. `$Year` and `$Month` are set by avhProcess based on the input file.

The user can override `$L3DataBase` since it is evaluated by the shell prior to use. For example, if the line:

```
L3DataBase="\$ApsDataBase/avhrr/\$Year"
```

is set in the areas script and we assume that $ApsDataBase is set to "/data" and that for a particular file $Year has been set to 1999, then the product file will be moved to /data/avhrr/1999. Note that to use the variables, the user must "escape" the '$' by inserting a "\".

`L4DAYDataBase`
: This variable is used to indicate the location of the Level-4 daily composites data base for the generated product file. By default, it is set to:

```
$ApsDataBase/$CompLevel/$Sensor/$Version/$Region/
daily/$Year/$Month
```

where, `$ApsDataBase` is defined in the `aps.conf` file and represents the top directory of the data base. `$CompLevel` is set to the string "lvl4" by `avhInit`. `$Sensor` is set to the string "avhrr" by `avhInit`. `$Version` is set to "5.0" by `avhInit`. `$Region` is set to `$MapName` by `avhInit`. `$Year` and `$Month` are set by `AvhProcess` based on the input file.

The user may override `$L4DAYDataBase` since it is evaluated by the shell prior to use.

`L4NDDataBase`
: This variable is used to indicate the location of the Level-4 weekly (8-day) composites data base for the generated product file. By default, it is set to:

  `$ApsDataBase/$CompLevel/$Sensor/$Version/$Region/weekly/$Year`.

  where, `$ApsDataBase` is defined in the `aps.conf` file and represents the top directory of the data base. `$CompLevel` is set to the string "lvl4" by `avhInit`. `$Sensor` is set to the string "avhrr" by `avhInit`. `$Version` is set to "5.0" by `avhInit`. `$Region` is set to `$MapName` by `avhInit`. `$Year` is set by `AvhProcess` based on the input file.

  The user may override `$L4NDDataBase` since it is evaluated by the shell prior to use.

`L4MODataBase`
: This variable is used to indicate the location of the Level-4 monthly composites data base for the generated product file. By default, it is set to:

  `$ApsDataBase/$CompLevel/$Sensor/$Version/$Region/monthly/$Year`.

  where, `$ApsDataBase` is defined in the `aps.conf` file and represents the top directory of the data base. `$CompLevel` is set to the string "lvl4" by `avhInit`. `$Sensor` is set to the string "avhrr" by `avhInit`. `$Version` is set to "5.0" by `avhInit`. `$Region` is set to `$MapName` by `avhInit`. `$Year` is set by `AvhProcess` based on the input file.

  The user may override `$L4MODataBase` since it is evaluated by the shell prior to use.

`L4YRDataBase`
: This variable is used to indicate the location of the Level-4 yearly composites data base for the generated product file. By default, it is set to:

  `$ApsDataBase/$CompLevel/$Sensor/$Version/$Region/yearly`.

  where, `$ApsDataBase` is defined in the `aps.conf` file and represents the top directory of the data base. `$CompLevel` is set to the string "lvl4" by `avhInit`. `$Sensor` is set to the string "avhrr" by `avhInit`. `$Version` is set to "5.0" by `avhInit`. `$Region` is set to `$MapName` by `avhInit`.

  The user may override `$L4YRDataBase` since it is evaluated by the shell prior to use.

`CmpOpt`
: This can be defined by the user to select the type of compression program to call for the output product file before it is moved to $L3DataBase. This option can be set to: "gzip", "compress", "bzip2" or "none". Only set CmpOpt to a compression type that is available on user's machine.

`Verbose`
: If defined this variable will cause the script functions to call 'set -x' within each script function. This will have the effect of printing out each step as it is executed.

`MapFile`
: Name of file containing image map file. Defaults to `$ApsData/maps.hdf`

`MapExt` This is a string that is appended to the Level-3 file which is written to the database. Usually it is a three character extension all uppercase.

`MinPixels`, `MinLines` Used to set the minimum pixels/lines that must be extracted from the AVHRR file by avhExtract to continue processing. These are used to insure that enough of the input file covers the area of interest. By default these are not defined and, therefore, no check is performed.

`AreaOpts` This allows the use of options to the avhArea(1) program to be added. However, this string sould not contain the -p, -l, or -M options.

Browse Image Variables

`L3BrowseList` This is a list of whitespace delimited products which are converted to browse images. The products in this list must also be present in the `$L3ProdList` variable. By default, no browse images are created. That is, BrowseList is not defined.

`L3BrowseDir` This variable is used to indicate the location of for the browse images. By default, it is set to: `$ApsImagBase/$Level/$Sensor/$Version/$Region/$Year/$Month` where, `$ApsImagBase` is set in the `aps.conf` file and represents the top directory of the browse data base. `$Level` is set to the string "lvl3" by `avhInit`. `$Sensor` is set to the string "avhrr" by `avhInit`. `$Version` is set to "5.0" by `avhInit`. `$Region` is set to `$MapName` by `avhInit`. $Year and $Month are set by avhProcess based on the input file.

The user can override $BrowseDataBase since it is evaluated by the shell prior to use. For example, if the line:

```
L3BrowseDir="\$ApsImagBase/browse/\$Year"
```

is set in the areas script and we assume that $ApsImagBase is set to "/data" and that for a particular file $Year has been set to 1999, then the browse image will be moved to /data/browse/1999.

`${prod}_BrowseScaling` With this option, the user's defines the desired output scaling for the browse images. The variable is formed by appending the term `_BrowseScaling` to the name of the product. For example, to produce an SST browse image that is 300 pixels by 400 lines and has a data range from 0.0 to 30.5 degrees, add the line:

```
sst_BrowseScaling="-r 0.0,30.5 -R 20,199 -s 300,400"
```

in the areas script. See imgBrowse(1) for more information on scaling parameters.

Program Variables

These variables define the programs used by avhScripts. The user can overide these to test a new version of a program. They are defined by `avhInit`.

`ApsInfo` Set to the name of the satellite specific program used to obtain information from input Level-1 file. Defaults to $ApsBin/avhInfo.

`AvhArea` Set to the name of the program used to determine if a AVHRR file covers a map. Defaults to $ApsBin/avhArea.

`AvhInfo` Set to the name of the program used to obtain information from AVHRR file. Defaults to $ApsBin/avhInfo.

`AvhClouds` Set to the name of the program used to produce clouds masks from input calibrated AVHRR images. Defaults to $ApsBin/avhClouds.

`AvhIngest` Set to the name of the program used to ingest and calibrate the AVHRR data. Defaults to $ApsBin/avhIngest.

`AvhSST` Set to the name of the program used to produce sea surface temperature estimates from input brightness temperature images. Defaults to $ApsBin/avhSST.

`AvhTurbid` Set to the name of the program used to produce visible data products from input percent albedo images. Defaults to $ApsBin/avhTurbid.

Part II. Command Line Reference

The chapters in Part II form a reference guide for each program available in the Automated Processing System for processing AVHRR data.

Name

avhArea — determines the file extents of a NESDIS Level-1B data file which covers an image map.

Synopsis

`avhArea [options] mapname file`

Description

Determines the file extents (start/stop pixel/line) of a NESDIS Level-1B file (still in sensor projection) that covers a map.

The command **avhArea** begins by reading in the map from the mapfile. If the file can not be opened or the named map is not in the file, a diagnostic is printed and the program will exit.

Next, the AVHRR file is opened and the navigation information initialized. If unable to open the file or get the navigation information from the file, the program will print a diagnostic and exit.

Once the navigation has been set, **avhArea** reads in every scan line and determines the latitude and longitude for 82 pixels over the entire scan line (or approximately every 25th pixel). For each point that falls within the desired maps, the starting and stopping sample (or column) number of the file is determined. The line extents are also determined by the first line that contains data that falls within the box and the last line that falls outside the box again. The file extents are adjusted to be slightly larger than those found by the above procedure to ensure that no data within the region is missed. These file extents will be printed to the screen. These are printed to stdout: starting pixel, space, ending pixel, space, starting line, space, ending line.

If the entire file covers the image map, then “Complete coverage” will be written to stdout. If no part of the file covers the image map, then “No coverage” will be written to stdout. Otherwise, the file extents (that is, “Partial coverage”) will be printed as above.

Based on the landmask, **avhArea** can also determine if any pixels within the region fell over water. If no samples fell over water then the message “No Water Coverage” is added. This can be used to determine if the file is to be processed even when it covers the interested area.

Options

-f type
: define the input file format (if program can not determine)

  1 = NESDIS Level-1B format
  2 = NESDIS KLM Level-1B format
  3 = NESDIS Level-0 format
  4 = NRL KLM Terascan HDF format

-l
: Don't output start/stop line locations

-L file
: Use file as the input land mask file. Defaults to `$APS_DATA/landmask.dat`

-M mapFile
: Use the given map file to find mapName. Defaults to `$APS_DATA/maps.hdf`

-n #
: Set the number of points across and down the image used to search for data coverage. The default is 82 points which yields a control point roughly every 25 pixels and lines. For small regions - that might "fall between the cracks" - this can be set to a higher number to create a finer grid.

-o type — Select the type of output. Currently, the only available option is `box` which will output the file extents in the format appropriate for the APS standard -B option. That is, `-B isp=isp,iep=iep,isl=isl,iepl=iepl`.

-p — Don't output start/stop pixel locations

-v — Verbose output

--help — Display program help.

--version — Display program name version and time of compilation.

Environmental Variables

APS_DATA The location of the APS data directory.

Examples

The examples below show the same input file run against two different geographical areas. The last examples shows the result of trying to use an invalid input.

Example 1. Determining Data Coverage using avhArea

```
$ avhArea GulfOfMexico noaa-14.970205200124.lac
1 1638 1021 2551
$ avhArea -p -M my_maps.hdf GulfOfMexico noaa-14.970205200124.lac
1021 2551
$ avhArea EastSea noaa-14.970205200124.lac
No coverage
$ avhArea Junk noaa-14.970205200124.lac
Map (Junk) does not exist in file ($APS_DATA/maps.hdf).
$ echo $?
1
$ export APS_ETC=/home/aps/aps_v3.8.2.3-72-g7b866d/etc
$ export APS_DATA=/home/aps/aps_v3.8.2.3-72-g7b866d/data
$ avhArea -o box -M $APS_ETC/maps.hdf MissBight \
          NSS.FRAC.M2.D07216.S1525.E1705.B0410506.SV
-B isp=683,iep=1441,isl=113,iel=483
```

Name

avhClouds — produces a bit image of tests used to determine cloud contaminated pixels

Synopsis

`avhClouds` *`file`*

Description

avhClouds will determine the cloud cover using several different tests based on the cloud detection algorithm of Saunder and Kriebel (1988). These tests require brightness temperatures from channels 3, 4, and 5. If the scene is a daytime scene, then the albedo from channels 1 and 2 are also required. Each test will set a bit

Note

Bit numbers presented here start with bit-1 and end with bit-32

in the output file. The tests are divided into day and night tests as well as land/sea/coast test.

The input file should be the output from avhIngest(1). That program produces the calibrated radiances and brightness temperature products as well as the `l2_flags` product used by this **avhClouds**.

These tests are performed on a 3x3 box with all edge pixels being marked as cloud automatically. The input file is read in and used to determine on a pixel by pixel basis some of the characteristics needed by the algorithm for that pixel.

If all the surrounding pixels are determined to be land, the center pixel is marked as land. If all the surrounding pixels are determined to be sea (not land), the center pixel is marked as sea. Otherwise, the center pixel is marked as coast. The program **avhIngest** uses a landmask file to set the LAND flag (bit 2).

For the day or night characteristic of a pixel, **avhClouds** will set all pixels to day or night, if upon initialization, the code determines that all pixels are day or night. This is determined by examining the Sun elevation for the four corner points and center of the input image. If all five points are defined as day, then all pixels are marked as day. If all four points are defined as night, then all pixels are marked as night. A Sun elevation greater than 15 degrees implies day and less than or equal to 15 implies night.

If this gross day/night check fails, then the DAY_TIME flag set in the `l2_flags` product in the input file by **avhIngest** will be consulted. In a similiar manner to land/sea/coast, the DAY_TIME flag (bit 3) for surrounding pixels is examined to determine if the center pixel is DAY or NIGHT. If all surrounding pixels are defined as DAY_TIME, then the center pixel is marked as DAY. If all surrounding pixels are defined as NIGHT_TIME, then the center pixel is marked as NIGHT.

Channel 4 Gross Cloud Check

If channel 4 temperature is too low, it is assumed that these are cloud-top temperatures. All values over sea that are less than 273.15 degrees Kelvin (0 degrees Celsius) are flagged as clouds. For land and coast pixels, 263.15 degrees Kelvin (-10 degrees Celsius) are used. Bit 17 (CLD_CH4_GROSS_CLOUD_CHECK) represents this test.

Channel 4 Spatial Coherence Test

The standard deviation in a 3x3 box surrounding the pixel in question for channel 4 is determined. If the standard deviation for sea (day or night) or land (night only) are greater than the thresholds (0.35 and 1.75), then the pixel is masked as cloudy. Bit 18 (CLD_CH4_SPATIAL_COHERENCE) represents this test.

Visible Channel Test

During the day, a high value of in channel 2 may indicate clouds cover. For land pixels, channel 1 is used for the test if available. The threshold of these land pixels is 40.0. For sea pixels, the visible threshold is 10. For coastal pixels, the visibile threshold is 15.0. Bit 19 (CLD_CH2_GROSS_CHECK) represents this test.

Channel 2 Spatial Coherence Test

For a day pixel, the standard deviation in a 3x3 box surrounding the pixel in question of channel 2 is determined. If the deviation for sea (day or night) are greater than 0.4, then the pixel is masked as cloudy. Bit 20 (CLD_CH2_SPATIAL_COHERENCE) represents this test.

NIR/VIS Test

During the day, the ratio of the NIR (channel 2) over VIS (channel 1) may indicate clouds. For land pixels, the default threshold is 0.0. For sea pixels the default threshold is 0.75. Bit 21 (CLD_NIR_VIS) represents this test.

Low Fog and Uniform Stratus Check

For a night pixel, the channel 4 - channel 3 difference greater than a given threhold (default 1.0) indicates low fog or uniform stratus clouds. For these pixels, bit 22 (CLD_LOW_FOG_UNIFORM_STRATUS) represents this test.

Medium and High Cloud Check

If the pixel in question is a night pixel, then the difference in ch3 - ch5 will indicate medium and/or high clouds. The default threshold for this value is 1.5. Bit 23 (CLD_MEDIUM_HIGH_CLOUD) represents this test.

Thin cirrus cloud check

If the pixel in question is a night pixel, then the difference in ch4 - ch5 is consulted against a 2-D table that will vary based on ch4 and the satellite zenith angle. If the channel 4-5 difference is greater than the found threshold, the pixel is flagged as clouds. Bit 24 (CLD_THIN_CIRRUS) respresents this test.

Options

-T file	This runs the program in `trace' mode.
-v	Verbose mode.
--help	Display program help.
--version	Display program name version and time of compilation.

Reference

Saunders, R.W. and Kriebel, K.T., 1988, *An improved method for detecting clear sky and cloudy radiances from AVHRR data* Int. Journal of Remote Sensing, 1988, Vol 9, No. 1, pp 123-150.

Name

avhDump — dumps AVHRR video data from a NESDIS Level-1B file.

Synopsis

`avhDump` -c *`n`* [-f *`1`*] *`filename`* *`chan.bin`*

Description

avhDump is used to dump AVHRR video data from a NESDIS Level-1B file. Currently, it only supports the dumping of AVHRR video data which must be selected with the -c option.

The output file is written as 16-bit integers in a flat binary format which has 2048 (LAC/HRPT) or 409 (GAC) columns by *n* number of rows. The number of columns and rows are printed by this program. Also, if the user knows the file type, the number of rows is therefore known and the number of columns can be computed by dividing the size of the file by 2 times the number of columns.

Options

-c n
: Select a channel to output. Must be between 1 and 5.

-f type
: define the input file format (if program can not determine)

  1 = NESDIS Level-1B format
  2 = NESDIS KLM Level-1B format
  3 = NESDIS Level-0 format
  4 = NRL KLM Terascan HDF format

-v
: Turn on verbose mode.

--help
: Display program help.

--version
: Display program name version and time of compilation.

Name

avhImage — creates a simple graphics image file from a NESDIS Level-1B file.

Synopsis

`avhImage` [*`options`*] *`filename`* *`image.ext`*

Description

avhImage is used to make a quick image from an AVHRR data file. By default, the program will read channel four counts (10-bits) and shift two bits to the right two for an output of the top 8 bits. If the input file is a LAC or HRPT file every fourth line and sample are written. If the input file is a GAC file, then every line/sample are written to the file.

The output file may be written as a PNM grayscale raw file, TIFF, PNG, or SGI RGB file depending on the compilation of the program. The available types may be optained by running **avhImage --help**. The type is selected based on the extension of the output file.

Options

-c n
: Select a channel to output. Must be between 1 and 5.

-f type
: define the input file format (if program can not determine)

  1 = NESDIS Level-1B format
  2 = NESDIS KLM Level-1B format
  3 = NESDIS Level-0 format
  4 = NRL KLM Terascan HDF format

-F
: Full output. If the input file is a LAC/HRPT file, this option writes the entire image to a file.

-t type
: Set the format type of the output image file. Valid responds are based on the compilation of the program. The *type* given is given as an image "extension". For example, *tiff*.

--help
: Display program help.

--version
: Display program name version and time of compilation.

Examples

To determine which formats are available, we run **avhImage** using the **--help** option.

```
$ avhImage --help

Usage: avhImage [OPTION] INPUT OUTPUT

Creates a subsampled grayscale image of
channel 4 from the AVHRR data stored in a
NESDIS Level-1B file, KLM Level-1B file, or
an HRPT Level-0 file.

OPTIONS
```

```
  -c n       use channel n for output
  -f type    set input file type
             1=NESDIS Level-1B file
             2=NESDIS KLM Level-1B file
             3=NESDIS Level-0 file
             4=TeraScan HDF file
  -F         do full output
  -t type    set output file type, see below
  --help     this output
  --version  version information

INPUT must be a one of the formats above
OUTPUT is a graphics file specified by its
extension.  It can one of the following:
  pnm       for PNM Format
  png       for PNG Format
  tiff      for TIFF Format
```

Since this version allows us to create PNG files, we create a quick-look PNG image of our input file using the following command:

```
$ avhImage NSS.HRPT.NL.D02142.S2044.E2056.B0858282.MO chan4.png
```

Name

avhInfo — queries information about a NESDIS Level-1B file(s).

Synopsis

`avhInfo` *`file`*...

`avhInfo` *`option`* *`file`*

Description

Run without options, **avhInfo** will write a report for each input file indicating satellite id, data type, etc. It may also be run with a single option and print the input file(s) value for that option. The first method is intended for interactive use at the shell prompt and the second method is intended for use within a shell program.

Options

-year	4-digit year of input file.
-doy	3-digit day of year of input file.
-month	3-character month of input file. Months are `jan', `feb', `mar', `apr', `may', `jun', `jul', `aug', `sep', `oct', `nov', `dec'
-time	6-digit time (HHMMSS) of input file.
-hour	2-digit hour (HHMMSS) of input file.
-min	2-digit min (MM) of input file.
-sec	2-digit second (SS) of input file.
-start_time	start time of input file.
-end_time	end time of input file.
-dsn	The NOAA defined Data Set Name from the TBM/ARS header.
-type	1-digit code for datatype, where: 1=LAC, 2=GAC, 3=HRPT
-sat	3-character satellite name. Names are `t-n', `n06', `n07', `n08', `n09', `n10', `n11', `n12', `n14', `n15', or `n16'.
-sat_code	3-character satellite name. Names are `TN', `NA', `NC', `NE', `NF', `NG', `ND', `NJ', `NK', or `NL'.
-name	Generate a file name in the following format as SSS.YYYY.MMDD.HHMM.T, where T is `l' for LAC, `h' is for HRPT and `g' is for GAC.
--help	Display program help.
--version	Display program name version and time of compilation.

Examples

Here is how a Bourne shell script function might use **avhInfo** to set the name of the output filenames:

Example 2. Extracting Information About AVHRR Dataset (programmatically)

```
set_name()
{
    sat=`avhInfo -sat $1`
    yr=`avhInfo -year $1`
    jday=`avhInfo -doy $1`
    time=`avhInfo -time $1`
    file=$sat.$yr$jday.$time.l1b
}
```

Here is an interactive use of **avhInfo**:

Example 3. Extracting Information About AVHRR Dataset

```
$ avhInfo  n17.2009114.0422.avhrr.hdf
Filename:         n17.2009114.0422.avhrr.hdf
Starting Time:    04/24/2009 04:22, 114
Ending Time:      unknown
Satellite:        noaa-17
File Type:        HDF
Data Set Name:    (null)
File Format:      TeraScan HDF
Datatype:         HRPT (High Resolution Picture Transmission)
Total Scans:      3759 with no gaps
Total Samples:    2048
Total Channels:   5   ( 1 2 3 4 5 )
Video Bits:       10 bits

$ avhInfo NSS.FRAC.M2.D07216.S1525.E1705.B0410506.SV
Filename:         NSS.FRAC.M2.D07216.S1525.E1705.B0410506.SV
Starting Time:    08/04/2007 15:25, 216
Ending Time:      08/04/2007 17:05, 216
Satellite:        metop-02
File Type:        NESDIS AVHRR 1b (FRAC)
Data Set Name:    NSS.FRAC.M2.D07216.S1525.E1705.B0410506.SV
File Format:      NESDIS KLM Level-1B (Version 5, 2007/116)
Datatype:         HRPT (High Resolution Picture Transmission)
Total Scans:      1748 with no gaps
Total Samples:    2048
Total Channels:   5   ( 1 2 3 4 5 )
Video Bits:       10 bits
```

Name

avhIngest — produces calibrated albedo and brightness temperature image from AVHRR data.

Synopsis

`avhIngest` *`avhrr.1b file`* [*`parameters`*...]

Description

This program reads the AVHRR data from a NESDIS Level-1B file, NESDIS KLM Level-1B, or NESDIS Level-0, or NRL TeraScan HDF Level-0 file and writes out calibrated data to an APS HDF file format. The albedo data is calibrated using the slope and intercept specified in the input file. The brightness temperatures are calibrated using an algorithm by Brown or by the calibration coefficients specified in the input file for KLM.

The user may select those parameters in the following table marked with a dagger on the command line. By default all of these parameters will be created unless unavailable due to the input.

Table 2. Ocean Parameters

Product	Description
l2_flags	Level-2 Processing flags
albedo_ch1	Percent albedo product for visible channel 1
albedo_ch1	Percent albedo product for visible channel 2
albedo_ch3	Percent albedo product for visible channel 3A
btemp_ch3	Brightness energy temperature (deg Kelvin) product for IR channel 3
btemp_ch4	Brightness energy temperature (deg Kelvin) product for IR channel 4
btemp_ch5	Brightness energy temperature (deg Kelvin) product for IR channel 5
latitudes	Latitudes
longitudes	Longitudes
solz	Solar zenith angles
sola	Solar azmiuth angles
senz	Sensor zenith angles
sena	Sensor azmiuth angles
secsat	Secant of the view angle

Note

Presently this software can only handle NOAA satellites 12, 14, 15, 16, 17, 18, 19 and MetOp-02.

The processing flags, `l2_flags` is a 32-bit product that indicates conditions for each pixel. The following table shows the bit location, name of the flag, and a short description of the flag.

Table 3. Level-2 Processing Flags

Bit	Flag	Description
1	NO_DATA	no video
2	LAND	Land mask
3	DAY_TIME	Pixel is considered day-time
4	NIGHT_TIME	Pixel is considered night-time
5	HIGLINT	High glint
6	HISATZEN	High satellite zenith
7	BATH	Coastal Water
9	FIRE	Possible Fire
13	HISOLZEN	High solar zenith angle
16	NO_GOOD	Data is outside of valid range (1E-05,1E+05)

NO_DATA - This flag indicates that the pixel was a fill pixel. These are pixels on the outside edges of the image and are based on the file extents used.

LAND - This flag indicates that the pixel was determined to be land using a landmask file. The default landmask file is the `$APS_DATA/common/landmask.dat`.

DAY_TIME - This flag indicates that the pixel had a solar zenith angle of less than 80.0 degrees. When the Sun is straight over-head the solar zenith angle is zero.

NIGHT_TIME - This flag indicates that the pixel had a solar zenith angle greater than 95.0 degrees.

HIGLINT - This flag indicates that the angles for this pixel are:.

HISATZEN - This flag indicates that the pixel had a satellite zenith angle greater than 55.0 degress.

FIRE - This flag indicates that the pixel passed steps 1 to 3 of the *Ionia File Project: AVHRR Active File Algorithm*.

HISOLZEN - This flag indicates that the pixel had a solar zenith angle greater than 75.0 degrees.

BATH - This flag indicates that the pixel was determined to have a water depth of less than 30 meters. This flag will not be defined if the bathymetry file was not found. The default bathymetry file is the `$APS_DATA/common/ETOP02.DOS`.

Options

-a angle
: If *angle* is defined then it is used to reduce the swath of the input image. This option can only be used for LAC/HRPT files and will calculate the number of pixels to reduce the image. It can be used to prevent the large pixels from the edge of the swath to be output. If angle is less than 1.1, then it is assumed to be given in radians. Otherwise it is give in degrees. A negative angle will be converted to a positive one.

-B isp iep isl iel isp iel irp irl
: These set up the subsection of the NESDIS Level-1B file to extract the data from. *isp* is the starting pixel number (1 to 2048). *iep* is the

	ending pixel number (1 to 2048, greater than isp). *isl* is the starting line (1 to *n*). *iel* is the ending line (1 to *n*, greater than isl). *irp* is the pixel subsampling factor. *irl* is the line subsampling factor.
-d	debug output
-f type	define the input file format (if program can not determine) 1 = NESDIS Level-1B format 2 = NESDIS KLM Level-1B format 3 = NESDIS Level-0 format 4 = NRL KLM Terascan HDF format
-L filename	use `filename` for land mask
-N	force day/night flag to night
-v	verbose output
--help	display program help
--version	Display program name version and time of compilation.

Examples

This example will produce floating point outputs of albedo channels 1 and 2 for input to the turbidity program **avhTurbid**.

Example 4. Ingest and Calibrate Albedo Channels

```
$ avhIngest n18.2009115.1906.avhrr.hdf avhrr.h5 albedo_ch1 albedo_ch2
$ hdf avhrr.h5 list
File  : avhrr.h5
Format: HDF v5

File Attributes: (43)

file = "test.h5"
fileTitle = "NRL Level-2 Data"
fileVersion = "3.2"
..... output removed
timeStartYear = 2009
timeStartDay = 115
timeStartTime = 68801262
timeStart = "Sat Apr 25 19:06:41 2009"
timeEndYear = 2009
prodList = "l2_flags,latitudes,longitudes,solz,sola,senz,sena,albedo
timeEndDay = 115
timeEndTime = 69641928
timeEnd = "Sat Apr 25 19:20:41 2009"
timeDayNight = "Day"
processedVersion = "5.0"
inputLevel1AFile = "n18.2009115.1906.avhrr.hdf"
..... output removed
albedo_ch1 [5036,2048]
    Albedo for channel 1
    in "% albedo" from 0 to 100
    based on the algorithm "NOAA Technical Memorandum NESS 107 - Rev
    images default to colortable nrl with range [1,100]
    stored as int16   using slope 0.002 and offset 0

albedo_ch2 [5036,2048]
    Albedo for channel 2
    in "% albedo" from 0 to 100
    based on the algorithm "NOAA Technical Memorandum NESS 107 - Rev
    images default to colortable nrl with range [1,100]
    stored as int16   using slope 0.002 and offset 0
..... output removed
$
```

Example 5. Ingest and Calibrate Thermal Channels for MissBight

```
$ avhArea -o box -M $APS_ETC/maps.hdf MissBight n18.2009115.1906.avhrr.hdf
-B isp=632,iep=1340,isl=2298,iel=2649
$ avhIngest -B isp=632,iep=1340,isl=2298,iel=2649 n18.2009115.1906.avhrr.hdf \
            therm.hdf btemp_ch4 btemp_ch4
```

References

Kidwell, K, 1999. *KLM User's Guide*. NCDC/NESDIS, National Climatic Data Center, Washington, D.C.

Kidwell, K, 1991. *NOAA Polar Orbiter User's Guide*. NCDC/NESDIS, National Climatic Data Center, Washington, D.C.

Brown, J. W., O. B. Brown, and R. H. Evans, 1993. *Calibration of AVHRR infrared channels: a new approach to non-linear correction.* J. Geophys. Res. 98 (NC10), 18257-18268.

Name

avhSST — produces sea surface temperature images from brightness temperature images

Synopsis

`avhSST file`

Description

This program will read brightness temperature data using channels 3, 4, and 5 produced by the program avhIngest(1) and generates an image file of sea surface temperature estimates. The algorithms available are based on the Multi-Channel Sea Surface Temperature (MCSST) and Non-Linear Sea Surface Temperature set of algorithms. Each set of algorithms is futher sub-divided by channel selection. For example, the MCSST Split algorithm uses AVHRR channels 4 and 5 (10 and 11 micron). The NLSST algorithm is considered non-linear since it includes some previous estimate of the sea surface temperature.

The set of algorithms is also sub-divided by day and night due to the effect of day-time heating. For each pixel, the **avhSST** will determine which algorithm (day or night) is used based upon the sun elevation. For each pixel where the sun elevation is less than 15 degrees, the night time algorithm is used. For all other pixels, the day-time algorithm is used.

MCSST Algorithms

The Multi-Channels algorithms consist of the split, dual, and triple.

Equation 1. MCSST Split

sst = a T4 + b (T4 - T5) - c (T4 - T5) sec(# - 1) + d

This equation is designed for AVHRR/2 and AVHRR/3 sensors which have both the 10 micron and 11 micron channels.

Table 4. MCSST Split Day Coefficients

Satellite	a	b	c	d
MetOp-02	1.0241	2.2458	0.8717	-280.0106
NOAA-19	1.028	1.7983	0.8069	-281.4467
NOAA-18	1.0232	2.0955	0.9460	-280.0467
NOAA-17	0.9894	2.5380	1.0108	-270.3393
NOAA-16	0.9836	2.2640	0.6999	-269.0536
NOAA-15	0.9824	2.5226	0.5445	-267.6418
NOAA-14	1.017342	2.139588	0.779706	-278.4300
NOAA-12	0.963563	2.579211	0.242598	-263.0060
NOAA-11	0.9781	2.3935	0.3098	-266.7104

Table 5. MCSST Split Night Coefficients

Satellite	a	b	c	d
MetOp-02	1.0095	2.4749	1.0438	-276.0721

Satellite	a	b	c	d
NOAA-19	1.0218	1.8459	0.8504	-279.7071
NOAA-18	1.0137	2.2144	0.8659	-277.5553
NOAA-17	0.9938	2.5765	1.0239	-271.7086
NOAA-16	0.9920	2.4204	0.7120	-271.7222
NOAA-15	0.9868	2.6895	0.6083	-269.3207
NOAA-14	1.029088	2.275385	0.752567	-282.2400
NOAA-12	0.967077	2.384376	0.480788	-267.3342
NOAA-11	0.9790	2.6072	0.6361	-267.5683

Equation 2. MCSST Dual

sst = a T3 + b (T3 - T4) - c sec(# - 1) + d

This equation is used only for night-time pixels. This equation is useful for AVHRR/1 sensor that do not have the 11 micron channel.

Table 6. MCSST Dual Coefficients

Satellite	a	b	c	d
MetOp-02	1.0092	0.4236	2.0028	-273.6964
NOAA-19	1.0385	0.3948	1.8968	-282.3408
NOAA-18	1.0230	0.3772	1.9748	-277.8950
NOAA-17	1.0230	0.4023	1.9881	-277.8805
NOAA-16	1.0134	0.5228	1.5947	-275.7492
NOAA-15	1.0332	0.5039	1.4596	-281.1191
NOAA-14	0.0	1.409936	1.975581	-273.9149
NOAA-12	0.0	1.288548	2.265075	-279.8460
NOAA-11	0.0	0.4721	1.9437	-272.1018

Equation 3. MCSST Triple

sst = a T4 + b (T3 - T5) - c sec(# - 1) + d

This equation is used only for night-time pixels. It is designed for AVHRR/2 and AVHRR/3 sensors which have both the 10 micron and 11 micron channels.

Table 7. MCSST Triple Coefficients

Satellite	a	b	c	d
MetOp-02	1.0039	0.9567	1.8621	-273.1243
NOAA-19	1.0205	0.8971	1.7604	-278.3586
NOAA-18	1.0140	0.9026	1.8639	-276.3201
NOAA-17	1.0077	0.9646	1.8702	-274.3670
NOAA-16	0.9973	0.9918	1.4523	-272.0867

Satellite	a	b	c	d
NOAA-15	1.0083	1.0400	1.1714	-274.6448
NOAA-14	1.010037	0.920822	1.760411	-275.3640
NOAA-12	1.000281	0.911173	1.710028	-271.9710
NOAA-11	0.9984	0.9916	1.5113	-271.9710

NLSST Algorithms

The non-linear sea surface temperature algorithms consist of the split, dual, and triple. As implemented by **avhSST** the *apriori* estimate of the sea surface temperature is first derived from the corresponding multi-channel sea surface temperature (MCSST) equation. The initial estimate from the MCSST is bounded at the top by 28 degrees Celcius.

Equation 4. NLSST Split

sst = a T4 + b (T4 - T5) - c sst (T4 - T5) sec(# - 1) + d

This equation is designed for AVHRR/2 and AVHRR/3 sensors which have both the 10 micron and 11 micron channels.

Table 8. NLSST Split Day Coefficients

Satellite	a	b	c	d
MetOp-02	0.9690	0.0772	1.0318	-263.3489
NOAA-19	0.9391	0.0652	0.7824	-254.9785
NOAA-18	0.9596	0.0699	0.9735	-260.7497
NOAA-17	0.9367	0.0848	1.0270	-254.2330
NOAA-16	0.9128	0.0803	0.7909	-247.7182
NOAA-15	0.9367	0.0864	0.5979	-253.8050
NOAA-14	0.939813	0.076066	0.801458	-255.1650
NOAA-12	0.876992	0.083132	0.349877	-236.6670
NOAA-11	0.9240	0.0827	0.4255	-250.3263

Table 9. NLSST Split Night Coefficients

Satellite	a	b	c	d
MetOp-02	0.9479	0.0838	1.0821	-257.3938
NOAA-19	0.9428	0.0669	0.8416	-256.1590
NOAA-18	0.9499	0.0722	0.8781	-258.0593
NOAA-17	0.9384	0.0857	1.0284	-254.7832
NOAA-16	0.9130	0.0808	0.7330	-247.7963
NOAA-15	0.9336	0.0930	0.6185	-253.1883
NOAA-14	0.933109	0.078095	0.738128	-253.4280
NOAA-12	0.888706	0.081646	57613628	-240.2290
NOAA-11	0.9063	0.0899	0.6716	-245.6227

Equation 5. NLSST Dual

sst = a T3 + b (T3 - T4) - c sec(# - 1) + d

This equation is used only for night-time pixels. This equation is useful for AVHRR/1 sensor that do not have the 11 micron channel.

Table 10. NLSST Dual Coefficients

Satellite	a	b	c	d
MetOp-02	1.0176	0.0145	2.0455	-276.1271
NOAA-19	1.0370	0.0150	1.8836	-281.9159
NOAA-18	1.0259	0.0132	1.9970	-278.7215
NOAA-17	1.0281	0.0138	2.0107	-279.3393
NOAA-16	1.0110	0.0183	1.6404	-274.9637
NOAA-15	1.0246	0.0196	1.4151	-278.5792
NOAA-14	1.019182	0.050086	2.039266	-276.8130
NOAA-12	1.0214680	0.050549	2.201377	-276.9000
NOAA-11	1.0096	0.0167	1.9319	-273.8171

Equation 6. NLSST Triple

sst = a T4 + b (T3 - T5) - c sec(# - 1) + d

This equation is used only for night-time pixels. It is designed for AVHRR/2 and AVHRR/3 sensors which have both the 10 micron and 11 micron channels.

Table 11. NLSST Triple Coefficients

Satellite	a	b	c	d
MetOp-02	0.9995	0.0329	2.0190	-271.5331
NOAA-19	0.9878	0.0317	1.7764	-268.4413
NOAA-18	0.9966	0.0305	1.9637	-270.8421
NOAA-17	0.9984	0.0329	1.9592	-271.3281
NOAA-16	0.9622	0.0336	1.6073	-261.3291
NOAA-15	0.9799	0.0364	1.1950	-266.0100
NOAA-14	0.980064	0.031889	1.817861	-266.1860
NOAA-12	0.963368	0.033139	1.731971	-260.8540
NOAA-11	0.9774	0.0342	1.6061	-265.1518

Default SST Equation

The following table show the default algorithm selected based upon the satellite identification.

Table 12. Default SST Equation Selection

Satellite	Day Algorithm	Night Algorithm
MetOp-02	NLSST Split (4/5)	NLSST Triple (3/4/5)

Satellite	Day Algorithm	Night Algorithm
NOAA-19	NLSST Split (4/5)	NLSST Triple (3/4/5)
NOAA-18	NLSST Split (4/5)	NLSST Triple (3/4/5)
NOAA-17	NLSST Split (4/5)	NLSST Triple (3/4/5)
NOAA-16	NLSST Split (4/5)	NLSST Triple (3/4/5)
NOAA-15	NLSST Split (4/5)	NLSST Triple (3/4/5)
NOAA-14	NLSST Split (4/5)	NLSST Triple (3/4/5)
NOAA-12	NLSST Split (4/5)	NLSST Split (4/5)
NOAA-11	NLSST Split (4/5)	NLSST Triple (3/4/5)

Processing Flags

The `l2_flags` product is updated by **avhSST** by appending a new flag `SST_FAIL`. This flag is used to indicate failure of the SST algorithm.

Table 13. Level-2 Processing Flags

Bit	Flag	Description
8	SST_FAIL	Sea Surface Temperature Algorithm Failure

SST_FAIL - This flag indicates that the sea surface temperature algorithm failed for the pixel.

Options

-C Sets the output temperature scale to Celsius

-d Force use of day algorithm

-e number Select a specific equation

-F Sets the output temperature scale to Fahrenheit

-n Force use of night algorithm

-p name Define the name of the output sst product (defaults to `sst`)

-T file This runs the program in `trace' mode

-v Verbose output

--help Display program help

--version Display program name version and time of compilation.

References

The coefficients for the various SST algorithms were obtained by NAVOCEANO.

Name

avhScan — dumps scan line information from a NESDIS Level-1B file

Synopsis

`avhScan` [-f *type*] [-n *num*] *file* [latlon | prt | solar | telm | thermal-cal | time | vis-cal]

Description

This program reads a NESDIS Level-1B file and writes to stdout information for each scan line based on the user specified target. The target can be one of the following: *latlon*, *prt*, *radiance*, *solar*, *telm*, *thermal-cal*, *time*, *vis-cal*.

If target *latlon* is selected, then **avhScan** will dump the scan line number, ascending/descending flag, and the number of valid lat/lon pairs followed by the 1st, 26th, and 51st lat/lon pair.

If *prt* is selected, then **avhScan** will dump the three prt counts stored in each scan line. Generally, these are duplicates of each other and contain the multiplexed PRT counts for each of the four PRTs. This dump, however, does not de-multiplex them.

If target *radiance* is selected, then **avhScan** will dump the scan line number followed by the radiance conversion values which include the central wave number and the two contants.

If target *solar* is selected, then **avhScan** will dump the scan line number, and the number of meaningful solar zenith angles followed by the 1st, 26th, and 51st solar zenith angle.

If the user selects *thermal-cal*, then the calibration values (slope/intercept) for all thermal channels will be dumped.

Target *telm* will cause **avhScan** to dump the the five ramp calibration counts, the three PRT counts, the ten internal target view counts for all three channels, and the ten space view counts for all five channels.

If the user selects *time*, then **avhScan** will dump the time embedded in each scan line.

If the user selects *vis-cal*, then the calibration values (slope/intercept) for all visible channels will be dumped.

Options

-f type	define the input file format (if program can not determine) 1 = NESDIS Level-1B format 2 = NESDIS KLM Level-1B format 3 = NESDIS Level-0 format 4 = NRL KLM Terascan HDF format
-n n	skip every nth record

Examples

Each of the examples below are from the file noaa-14.970205200124.lac. They do not represent the full output of the program, but an excert to show the format used.

Example 6. Scanning for Latitude/Longitude

```
$ avhScan -n 50 noaa-14.970205200124.lac latlon
    1      1 051 0    8.56/ -77.41    6.76/ -89.91    4.64/-102.32
   51     51 000 0    8.56/ -77.41    6.76/ -89.91    4.64/-102.32
  101    101 051 0    9.79/ -77.66    8.01/ -90.20    5.85/-102.63
  151    151 051 0   10.30/ -77.76    8.54/ -90.32    6.37/-102.77
  201    201 051 0   10.79/ -77.85    9.03/ -90.43    6.85/-102.89
```

Example 7. Scanning for PRT counts

```
$ avhScan noaa-14.970205200124.lac prt | more
    1     1      917    939      3
    2     2      917    683      3
    3     3      917    939      3
    4     4      917    811      3
    5     5      917    939      3
    6     6      917    811      3
```

Example 8. Scanning for solar angles

```
$ avhScan -n20  noaa-14.970205200124.lac solar | more
    1      1  51     46.50   35.50   25.50
   21     21  51     47.00   36.00   26.00
   41     41   0     47.00   36.00   26.00
   61     61  51     47.00   36.00   26.50
   81     81  51     47.00   36.00   26.50
  101    101  51     47.00   36.50   26.50
  121    121  51     47.50   36.50   26.50
  141    141  51     47.50   36.50   27.00
  161    161  51     47.50   36.50   27.00
  181    181  51     47.50   36.50   27.00
  201    201  51     47.50   37.00   27.50
  221    221  51     47.50   37.00   27.50
  241    241  51     48.00   37.00   27.50
```

Example 9. Scanning for Telemetry Data

```
$ avhScan -n20  noaa-14.970205200124.lac telm | more
Record/Line     1/    1
Ramp           644     367     860     413     527
TrgTemp        917     939       3
Black Body
     72      72     617     617     738     740     736     740     733     737
    708     708     119     119     391     391     391     423     423     391
    862     862     120     120     371     370     370     371     371     371
Space
    734     371     736     987     982     987     977     989      41      41
    391       0     391     480     488     993     992       0     985     988
    371     738     372    1021     989     989     733     864     992     993
    737     399      41      41     169      41      41     989     990     990
    391     371     169      41      41     169     169      45      41     169
-----------
Record/Line    21/   21
Ramp           644     367     860     413     527
TrgTemp        917     939       3
Black Body
     72      72     266     266     226     738     733     737     724     735
    708     708     161     161     391     391     391     391     391     391
    869     869     161     161     371     355     370     370     371     371
Space
    733     371     736     988     984     989     986     991      41      41
    391       0     391     992     992     992     992       0     982     990
    371     739     371     477     989     990     989     992     736     868
    736     391      41      41      41      41      41     989     989     989
    391     371      41      41      41      41     169      41      57      41
-----------
Record/Line    41/   41
```

Example 10. Scanning Time Information

```
$ avhScan -n20  noaa-14.970205200124.lac time | more
      1   00001   1997 036 72186473  02/05/1997 20:03:06.473
     21   00021   1997 036 72193290  02/05/1997 20:03:13.290
     41   00041   1997 036 72197306  02/05/1997 20:03:17.306
     61   00061   1997 036 72201140  02/05/1997 20:03:21.140
     81   00081   1997 036 72204473  02/05/1997 20:03:24.473
    101   00101   1997 036 72207806  02/05/1997 20:03:27.806
```

Name

avhSwapL0 — converts a Terascan Level-0 formatted file to NESDIS Level-0 formatted file

Synopsis

`avhSwapL0` <*input.l0*> <*output.l0*>

Description

This program will take an AVHRR "hrpt" Terascan Level-0 formatted file and convert it to a NESDIS Level-0 format required for input to avhL1bgen. The Terascan Level-0 format can be created by the **archive** command of the Terascan system. The input file consists of 44 512-byte sectors in which the 10-bit AVHRR Level-0 stream in merged into 16-bits. This program will swap the bytes to that they resemble the HPRT Minor Frame format discribed in NOAA Technical Memorandum NESS 107 - Rev 1.

Name

avhTurbid — produces beam attenuation, diffuse attenuation, suspended solids images from percent albedo images.

Synopsis

`avhTurbid` [-g *angle*] [-n *angle*] *file*

Description

This program is used to estimate the beam attenuation coefficient (c_660), the diffuse attenuation coefficient for photosynthetically active radiation for 400-700 nm (KPAR), and total suspended solids (or seston) in coastal waters using the percent albedo for channels 1 and 2 of the AVHRR data. It implements the algorithm of Rick Stumpf to perform the atmospheric correction.

The input file must contain the `albedo_ch1` and The products produced by this program will be appended to the input file. The percent albedo channels are products of the program avhIngest(1).

Options

-g angle
: This sets the glint angle used to determine if an area is susectible to glint. If an area is found to be glint contaminated, it is masked out. By default, this angle is set to -1.0 to indicate that no glint masking is being performed.

-n angle
: This is the angle used to determine if for the selected control point, the sun is too low or down. If the solar zenith angle is greater than this angle the control is not processed. By default the angle is set to -1.0.

--help
: Display program help.

--version
: Display program name version and time of compilation.

Examples

This example shows how to generate the turbidity products from an AVHRR day-time data file (an NRL Terascan/HDF) using avhIngest(1) and **avhTurbid**.

Example 11. Computing Turbidity Products

```
$ export APS_DATA=$APS_DIR/data   # required by avhIngest
$ avhIngest n18.2009115.1906.avhrr.hdf avhrr.h5 albedo_ch1 albedo_ch2
$ avhTurbid avhrr.h5
```

References

Stumpf, R. P. (1992) *Remote Sensing of Water Clarity and Suspended Sediments in Coastal Waters*. In Proceedings of the First Thematic Conference on Remote Sensing for Marine and Coastal Environments. SPIE 1930, 293-305. Environmental Research Institute of Michigan, Ann Arbor, MI.

Gould, R. W. and Arnone, R. A. (1997) *Estimating the beam attenuation coefficient in coastal waters from AVHRR imagery*. Continental Shelf Research, Vol 17, No 11, p 1375-1387.